SUN/226.13

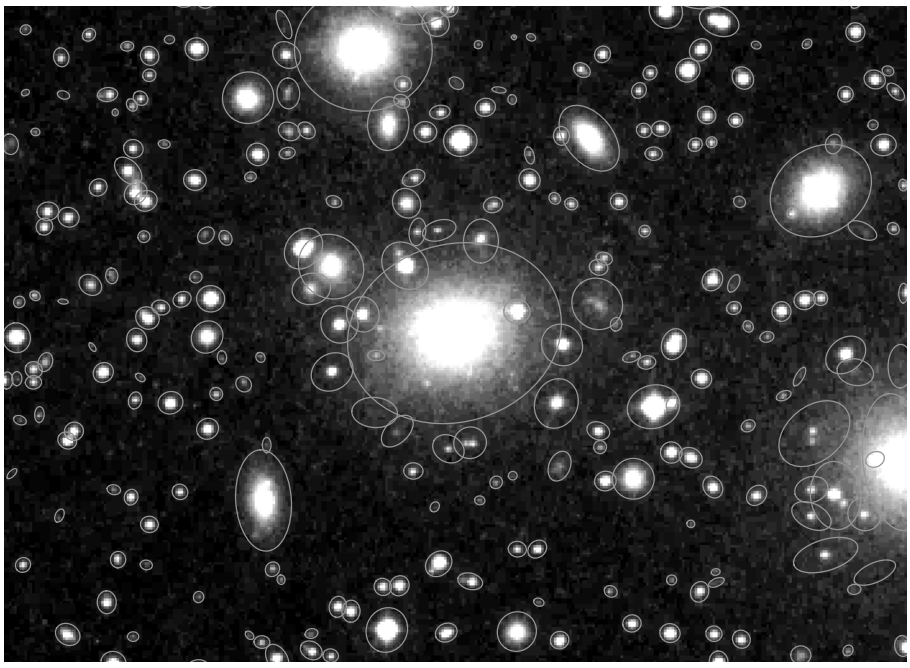
Starlink Project
Starlink User Note 226.13

A.J. Chipperfield
P.W. Draper

19th September 2014

EXTRACTOR
An Astronomical Source Detection Program

User Manual



Abstract

EXTRACTOR will detect sources in an astronomical image and build a catalogue listing them. It is based on the popular SExtractor program, has very flexible configuration facilities and can handle images and catalogues in a variety of formats.

Contents

1 Introduction **1**

2 Running EXTRACTOR **1**
2.1 Running EXTRACTOR from scripts 2

3 Using images in non-NDF formats **3**

4 Running SExtractor **4**

5 Using GAIA and EXTRACTOR **4**

6 Implementation notes **4**

7 Isophotal radii **5**

8 References **6**

A Specification of EXTRACTOR **7**
EXTRACTOR 8

1 Introduction

EXTRACTOR is a program for automatically detecting objects on an astronomical image and building a catalogue of their properties. It is particularly suited for the reduction of large scale galaxy-survey data, but also performs well on other astronomical images.

EXTRACTOR is Emmanuel Bertin's SExtractor (Source-Extractor) program[1] re-packaged for use in the Starlink Software Environment (see SG/4). This means that it uses the Starlink parameter system, accepts images in NDF format (see SUN/33) and uses the AST library (SUN/210) for astrometry.

A slightly modified version of SExtractor 2.19.5 is also included in the distribution of EXTRACTOR. You can find details of SExtractor's operation, and how to configure it, in the SExtractor User's Guide (which is issued as a Starlink Miscellaneous User Document, MUD 165[2]) – you configure EXTRACTOR in much the same way, although with some restrictions (see Section 6). The SExtractor home page – http://terapix.iap.fr/soft/sextractor – offers more information, such as a link to the *SExtractor for Dummies*.

In the Starlink release both EXTRACTOR and SExtractor do not offer the galaxy model fitting features that the full SExtractor builds do. This has been done to reduce the external package dependencies required, also it isn't clear how useful these abilities would be without also creating an NDF version of PSFEx.

This document describes how to run EXTRACTOR, and includes instructions for running SExtractor from the Starlink distribution.

2 Running EXTRACTOR

Like any Starlink program EXTRACTOR may be run from the Unix shell or from a user-interface such as ICL (see SG/5). A useful summary of the rich variety of methods of specifying parameter values is given in SUN/95. C-shell users should also consult SC/4, the C-Shell CookBook.

The examples below assume you are running from the Unix shell but the commands and parameters are exactly the same from ICL.

After normal Starlink startup, at the shell prompt, type:

```
% extractorsetup
```

to initialise the package and then:

```
% extractor
```

to run the program. You will now be prompted for the values of parameters CONFIG and IMAGE in turn:

[1] BERTIN E. and ARNOUTS S., 1996, A&AS 117,393

[2] Note this document is still in draft, consequently you should read this in conjunction with the A&AS paper and the earlier SExtractor user document.

```
CONFIG - Configuration file /'$EXTRACTOR_DIR/config/default.sex'/ >
IMAGE - Input image /'image'/ >
```

where CONFIG is the name of the 'preferences' file and IMAGE is the name of the image file to be processed.

Suggested values are given between // in the prompt. You can accept the suggested values by just typing <RETURN>. If you accept the suggested values shown above, EXTRACTOR will process the NDF "image" using the installed default configuration files to produce a catalogue named test.cat (this name is specified in the preferences file).

CONFIG has a 'default' value of $EXTRACTOR_DIR/config/default.sex. This file has some sensible defaults but you will probably want to copy it to your own directories and modify it to your taste (as you would for the native SExtractor program). Similarly with the other configuration files. The environment variable EXTRACTOR_DIR is defined to point to the directory containing the EXTRACTOR program and the config directory.

You can also provide parameter values on the command line. Either positionally,

```
% extractor image $EXTRACTOR_DIR/config/default.sex
```

or by keyword,

```
% extractor config=$EXTRACTOR_DIR/config/default.sex
```

(In this case, you would be prompted for IMAGE but not for CONFIG.)

The additional parameters KEYWORDS, NAME and VALUE allow configuration parameters specified in the preferences file to be overridden without editing the file. For example:

```
% extractor keywords=true
NAME - Parameter name /!/ > catalog_name
VALUE - Parameter value /!/ > sky.cat
NAME - Parameter name /!/ > catalog_type
VALUE - Parameter value /!/ > ascii_skycat
NAME - Parameter name /!/ >
CONFIG - Configuration file /'$EXTRACTOR_DIR/config/default.sex'/ >
IMAGE - Input image /'image'/ >
```

would change the name and type of the catalogue produced (for this run only). The list of changes is terminated by replying with the NULL response (!) to the prompt. The SExtractor User's Guide (MUD/165) gives a full list of possible parameter names and values, but it is only sensible to change a few in this way.

2.1 Running EXTRACTOR from scripts

If you'd like to run EXTRACTOR from a script, just say for instance sampling different populations of objects at different thresholds, you can do this using the NAME VALUE parameters (avoiding the need to have multiple configuration files), but you need to adopt a slightly different strategy to normal programs. Here's one example script:

Example batch script

```
#!/bin/csh

#  Initialize EXTRACTOR
extractorsetup

#  Extract all objects above 1 sigma
extractor keywords=true image=image config=default.sex <<EOF
catalog_name
thresh1.cat
detect_thresh
1.0
!
EOF

# Extract all objects above 2 sigma and measure on a different image.
extractor keywords image='"detect,measure"' config=default.sex <<EOF
catalog_name
thresh2.cat
detect_thresh
2.0
!
EOF
```

Using the C-shell <<EOF mechanism allows you to send information to the program as if you're typing it in.

3 Using images in non-NDF formats

If you want to read an image in one of the formats handled by the CONVERT package, just start up CONVERT before running EXTRACTOR and specify the image with an appropriate extension. For example:

```
% convert

   CONVERT commands are now available

   Defaults for automatic NDF conversion are set.

   Type conhelp for help on CONVERT commands.
   Type "showme sun55" to browse the hypertext documentation.

% extractor
CONFIG - Configuration file /'$EXTRACTOR_DIR/config/default.sex'/ >
IMAGE - Input image /'image'/ > image.fit
```

will use the FITS file image.fit, automatically converting it to a temporary NDF file and then reading the temporary file.

4 Running SExtractor

A native, FITS-only, version of SExtractor is also distributed with the EXTRACTOR package for convenience and use in the object detection toolbox in the GAIA application. The executable image is installed in $EXTRACTOR_DIR and is ran as described in the SExtractor User's Guide(MUD/165).

To use this version either put $EXTRACTOR_DIR on your PATH or define an alias:

```
% alias sex $EXTRACTOR_DIR/sex
```

then type:

```
% sex image [-c configuration-file] [-Parameter1 Value1] [-Parameter2 Value2]
...
```

Note that if the -c option is omitted, file default.sex in the current working directory will be used – if you want to use the installed default file, you will need to specify it explicitly.

5 Using GAIA and EXTRACTOR

The Graphical Astronomy and Image Analysis Tool (GAIA - SUN/214), has an interactive toolbox facility that uses the EXTRACTOR (and SExtractor) programs. This allows you to interactively adjust the detection preferences and identifies the objects that have been detected by drawing suitable ellipses over your image (in fact GAIA produced the image used on the title page of this document).

GAIA also displays the results catalogue so you can view the measurements associated with each object and vice versa (*i.e.* you can select one of the displayed ellipses and view the associated measurements, or select a row of measurements and view the associated object). The catalogue interface also allows you to sort and select objects on the basis of their measurements.

6 Implementation notes

EXTRACTOR, will read data in floating point format. It will also read NDFs that contain BAD pixels. These will appear as "missing" regions, so any objects that are intersected will be split into parts.

Since EXTRACTOR uses the AST (SUN/211) library to perform astrometry, it can be used with images that contain DSS calibrations, as well as FITS-WCS and AST native ones (native SExtractor will only use images that have FITS-WCS calibrations). The GAIA display tool is a convenient way to add such astrometrical calibrations.

Two current limitations are that no support is provided for the use of NDF variance arrays (although these would probably map onto the "weight" image concept of SExtractor) and

that the propagation of the input NDF to the check image is not done (so WCS calibration, for instance, is lost).

NDF pixel coordinates (*i.e.* ones that include the NDF origin) can be obtained using the X_PIXEL and Y_PIXEL parameters.

The SExtractor program has a very large range of processing options. Consequently many of these options have not been tried extensively in the EXTRACTOR incarnation, so some problems may arise, when using non-default configurations.

A SExtractor facility that has not been implemented is the piping of catalogues to standard output. The use of weight and flag images are not supported.

Finally as noted earlier the galaxy fitting parameters are not available.

If you come across any problems with EXTRACTOR please notify Starlink Software Support (starlink@jiscmail.ac.uk).

7 Isophotal radii

SExtractor can measure the areas of objects at 8 optimally selected thresholds (such information is often used to estimate object profiles). These thresholds are consequently different for each object. Traditionally this has not been the case and isophotal areas/radii have been measured at fixed magnitude intervals above the detection threshold. Consequently, for convenience and compatibility with existing data and analysis methods, the EXTRACTOR and Starlink SExtractor programs have been extended to provide a flexible configuration scheme that allows this traditional behaviour to be recovered and naturally extended.

The configuration options that control the chosen isophotal thresholds are:

- RAD_TYPE and
- RAD_THRESH

RAD_TYPE can be either SB or INT, which indicate that the levels will be defined in terms of surface brightnesses or intensities (*i.e.* magnitudes and data counts), respectively.

The RAD_THRESH option can have up to three qualifying values, depending on the value of RAD_TYPE. If RAD_TYPE is SB then you should enter a line consisting of:

```
RAD_THRESH       step[,start,zp]
```

in your default.sex file. The value step being the required interval between levels, start being the value used as the first threshold and zp the data zero point, all in magnitudes per square arc-second. If only one value is given then the starting point is assumed to be the analysis threshold and the zero point is derived from the photometric value (MAG_ZEROPOINT).

If a RAD_THRESH value is not given then a default step of 0.75 magnitudes per square arcsec is used.

The actual formula used to generate the thresholds is:

$$I_i = A * 10^{-0.4*(start+step*i-zp)},\ i = 0,15$$

where I_i is the ith threshold, A is the area of an image pixel in arcseconds, $start$, $step$ and zp are as described above.

If `RAD_TYPE` is `INT` then the correct `RAD_THRESH` format is:

```
RAD_THRESH          step[,start]
```

`step` being the interval between thresholds in magnitudes and `start` being the threshold used for the first level. If `start` is not given then the analysis threshold is used.

The formula used to generate these thresholds is:

$$I_i = start * 10^{-0.4*step*i},\ i = 0,15$$

If no `RAD_THRESH` values are given this time then the APM/PISA (see SUN/109) analysis thresholds are used:

$$I_i = start * 2^{(i+2)},\ i = 1,15$$

This gives approx 0.75 magnitude steps ($2.5 * log(2)$). The first threshold is set to the analysis threshold.

Notes: before any measurements will be made at least one of the catalogue parameters `RAD0` through `RAD15` must be present in the `default.param` file. The minimum starting threshold that can be used is the analysis one — no information about values below this is available.

For completeness the existing `ISO0-ISO7` areas in `SExtractor` are based on an `optimal` sampling of each object profile. Under this scheme each threshold is:

$$I_i = start * (\frac{I_p}{start})^{i/8},\ i = 0,7$$

where I_p = peak intensity. So you get a range of levels for each object spanning the range from its analysis threshold to just below the peak intensity.

8 References

Bailey, J.A. : SG/5 : ICL – The Interactive Command Language for ADAM.
Bertin, E. : MUD/165 : SExtractor User's Guide (DRAFT).
Bertin, E. and Arnouts, S., 1996, A&AS 117,393
Currie, M.J. & Berry D.S.: SUN/95 : KAPPA – Kernel Application Package.
Currie, M.J. *et al.* : SUN/55 : CONVERT – A Format-conversion Package.
Davenhall, A.C. : SUN/190 : CURSA – Catalogue and Table Manipulation Applications.
Draper, P.W. & Gray, N.: SUN/214 : GAIA – Graphical Astronomy and Image Analysis Tool.
Lawden, M.D.: SG/4 : ADAM – The Starlink Software Environment.
Warren-Smith, R.F. : SSN/20 : Adding Format Conversion Facilities to the NDF Data Access Library.
Warren-Smith, R.F. : SUN/33 : NDF – Routines for Accessing the Extensible N-Dimensional Data Format.

A Specification of EXTRACTOR

EXTRACTOR
Extracts sources from astronomical images.

Description:

EXTRACTOR is a program for detecting and measuring the properties of all the sources on an astronomical image. It offers a large range of configuration options for controlling the way that objects are detected and the measurements that are made of them.

The source measurements are written to a catalogue (which can be of several different formats), so that they can be analysed (possibly by a catalogue handling package like CURSA – SUN/190).

EXTRACTOR, is based on the SExtractor program which is described in the SExtractor's User Guide (MUD/165). Consult this about all the various options that are available and for the rationale behind the program.

Usage:

```
extract image config [keywords] [name] [value]
```

Parameters:

IMAGE = LITERAL (Read)

The name of the image which contains the objects you wanted detected and parameterised. If you have initialised the CONVERT package (see SUN/55) then you may process foreign formats, such as FITS and IRAF.

Using this parameter you may give two image files. The first image will be used for detection and parameterising and the second will be used to actually measure the data values. Using this method allows you to measure the same objects many images, or to use a high signal to noise image to determine the measurement regions on a low signal to noise image

[global_data_file]

CONFIG = LITERAL (Read)

The name of the file that contains the many program parameters (things like the threshold for object detection). This is initially a file named default.sex that can be found in the directory $EXTRACTOR_DIR/config. To modify the parameters used by this program, you must take a copy of this file and edit it. Guidance about the values that parameters can take may be found in this file as well as in the associated SExtractor documentation (see MUD/165).

The measurements made are determined by a list of parameters in the file $EXTRACTOR_DIR/config/default.param. Again if you want measurements that are not available by default, you must take a copy of this file and edit it. Remember to also change default.sex to use this file (otherwise you will continue to use the system-wide defaults).

One off modifications of parameters can be made using the KEYWORDS, NAME and VALUE parameters.

[$EXTRACTOR_DIR/config/default.sex]

KEYWORDS = _LOGICAL (Read)

Whether you want to enter a series of parameter names and values interactively. If TRUE then the parameters NAME and VALUE are used to cyclically prompt for program parameters and the values you want to use. To end the cycle respond with a null symbol (!)

[FALSE]

NAME = LITERAL (Read)

The name of a preferences parameter that you want to set interactively. Respond with **!** when you have no more to enter. [!]

VALUE = LITERAL (Read)

The value of the parameter you have just specified using the NAME prompt. [!]